UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/614,363 | 07/12/2000 | John M. Airey | 15-4-632.51 | 2211 |

28393 7590 04/18/2006

STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C.
1100 NEW YORK AVE., N.W.
WASHINGTON, DC 20005

| EXAMINER | |
|---|---|
| WANG, JIN CHENG | |
| **ART UNIT** | **PAPER NUMBER** |
| 2628 | |

DATE MAILED: 04/18/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/614,363 | AIREY ET AL. | |
| | Examiner | Art Unit | |
| | Jin-Cheng Wang | 2628 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *15 March 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-3,5-13,22,26-33,35-37 and 45-56* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3,5-13,22,26-33,35-37 and 45-56* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

### *Response to Amendments*

Applicant's submissions filed on 3/15/2006 have been entered. Claims 4, 14-21, 23-25, 34, 38-44 have been canceled. Claims 1-3, 5-13, 22, 26-33, 35-37, and 45-56 are pending in the application.

### *Response to Arguments*

Applicant's arguments filed March 3, 2006 have been fully considered but are moot in view of the new ground(s) of rejection set forth in the present Office Action.

Rossin teaches a rasterization circuit coupled to the processor that rasterizes the primitive according to a rasterization process which operates using a floating point format (col. 7, lines 18-41; col. 3. lines 1-19), a frame buffer coupled to the rasterization circuit for storing a plurality of image values and a display screen coupled to the frame buffer for displaying an image according to the image values stored in the frame buffer (col. 2, lines 12-41. col. 3. lines 20-32).

In other words, Rossin teaches a typical computer graphics system include a geometry accelerator, a rasterizer and a frame buffer. The output from the geometry accelerator, referred to as rendering data, is used by the rasterizer (and optional texture mapping hardware) to compute final screen space coordinates and R, G, B color values for each pixel constituting the primitives. The pixel data is stored in the frame buffer for display on a display screen. In that the geometry accelerator may be required to perform on the order of hundreds of millions of floating point calculations per second per chip. Functions of the geometry accelerator may include three-dimensional transformation, lighting, clipping, and perspective divide operations as well as plane

equation generation, performed in floating point format. Geometry accelerator functions result in rendering data which is sent to the frame buffer subsystem for rasterization, and thereby the rasterization process which operates using a floating point format.

Rossin fails to expressly teach "a floating point frame buffer" or "a display screen coupled to the frame buffer for receiving the plurality of image values read out from the frame buffer in the floating point format." Rossin fails to explicitly teach a processor for performing geometric calculations on a plurality of vertices of a primitive.

Deering teaches a floating point frame buffer such as the frame buffer 100 for storing floating point z-values (See Deering **column 14, lines 15-20** and column 16, lines 16-30) and "a display screen coupled to the frame buffer for receiving the plurality of image values read out from the frame buffer in the floating point format wherein Deering teaches rendering the floating-point z-values from the floating point frame buffer 100 (See Deering column 14, lines 15-20 and **column 17, lines 10-20**) and Deering teaches that the floating point z-values representing z coordinates for vertices of triangle primitives usable **to render** three-dimensional objects on display device 84 (column 10, lines 60-67 and column 11, lines 1-5).

Deering additionally discloses the claim limitation of a processor for performing geometric calculations on a plurality of vertices of a primitive wherein Deering teaches the floating point processors 152A-152F of Fig. 6 including the F-core block 202 for performing the floating point intensive operations (which performs fixed point calculations as well) including the geometry transformation, clip testing, face determination, perspective division and screen space conversion (See column 7, lines 50-67).

Deering additionally discloses the claim limitation of a rasterization circuit coupled to the processor that rasterizes the primitive according to a rasterization process which operates using a floating point format wherein Deering teaches the draw processor 172A and 172B rasterizes the primitive (See column 9, lines 50-67 and column 10, lines 1-50) wherein the rasterization process includes partitioning primitives into pieces, sending a line description to the span field unit, performing an interpolation of values across arbitrarily oriented spans for triangles and lines, performing blending, anti-aliasing, depth cueing and setup for logical operations, etc. See Deering column 9, lines 50-67 and column 10, lines 1-50. The drawing processors 172A and 172B outputs floating point z-values to the floating point frame buffer 100 (See **column 14, lines 13-24**). Therefore, Deering's draw processor operates using a floating point format.

Therefore, having the combined teaching of Rossin, Deering and Olano as a whole, one of ordinary skill in the art would have found it obvious to modify the frame buffer of Rossin to achieve floating point precision (Olano Page 70) wherein *the floating point frame buffer generates a marked improvement in the rendered image quality* (Olano Fig. 4.7 and Page 59 and **Deering column 4, lines 25-30**) with more expensive computation load (Olano Page 69).

Moreover, Olano teaches a pixel processor for performing geometric calculations on a plurality of vertices of a primitive (See Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 68-75, 102-104). For example, Olano teaches pixel processor receives geometry primitive data and performs either floating point or fixed point operations on the received geometry data. He discloses a graphics rendering pipeline mapped to the so called PixelFlow including a SIMD system of pixel processors performing modeling, transformation, primitive and interpolation, shading, lighting,

atmospheric shading and image warping. In that the PixelFlow includes shaders for determining the shading and color variations across each surface wherein the shaders are executed sequentially including performing the surface shader to perform a certain class of texture lookups in which detailed surface geometry may be rendered using texture maps wherein the texture maps (Olano Page 31-32) are used to get different effects. In that he also teaches handling a floating point or fixed point frame buffer which is a portion of the rasterization pipeline within the graphics rendering pipeline. The color values received by the pipeline are represented in a floating point format which includes a mantissa portion and an exponent portion (Olano Page 100).

Fianlly, having the combined teaching of Rossin, Deering and Olano as a whole, one of ordinary skill in the art would have found it obvious to modify the rasterization process of Rossin which acts on the floating point color values that incorporates a processor in a graphics pipeline of Deering or Olano for performing geometric calculations on a plurality of vertices of a primitive. Doing so would enable the color values being represented more efficiently resulting in increased performance and accuracy (See Olano Fig. 4.7 and **Deering column 4, lines 25-30**) for the graphics pipeline (See Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 59, 68-79, 102-104) wherein the pipeline executing on the floating point values generates a marked improvement in the rendered image quality over the pipeline executing on the fixed point values (Olano Fig. 4.7 and Page 59 and **Deering column 4, lines 25-30**) with more expensive computation load (Olano Page 69).

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3, 5-13, 22, 26-33, 35-37, and 45-56 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rossin et al. (US Patent No. 5,862,066) in view of **Deering U.S. Patent No. 6,115,047 (hereinafter Deering)** and Marc Olano, "A Programmable Pipeline for Graphics Hardware", PhD Dissertation, Department of Computer Science, University of North Carolina, Chapel Hill, April 1998 (hereinafter Olano).

Re claims 1 and 45, Rossin teaches a rasterization circuit coupled to the processor that rasterizes the primitive according to a rasterization process which operates using a floating point format (col. 7, lines 18-41; col. 3. lines 1-19), a frame buffer coupled to the rasterization circuit for storing a plurality of image values and a display screen coupled to the frame buffer for displaying an image according to the image values stored in the frame buffer (col. 2, lines 12-41. col. 3. lines 20-32).

In other words, Rossin teaches a typical computer graphics system include a geometry accelerator, a rasterizer and a frame buffer. The output from the geometry accelerator, referred to as rendering data, is used by the rasterizer (and optional texture mapping hardware) to compute final screen space coordinates and R, G, B color values for each pixel constituting the primitives.

The pixel data is stored in the frame buffer for display on a display screen. In that the geometry accelerator may be required to perform on the order of hundreds of millions of floating point calculations per second per chip. Functions of the geometry accelerator may include three-dimensional transformation, lighting, clipping, and perspective divide operations as well as plane equation generation, performed in floating point format. Geometry accelerator functions result in rendering data which is sent to the frame buffer subsystem for rasterization, and thereby the rasterization process which operates using a floating point format.

Rossin fails to expressly teach "a floating point frame buffer" or "a display screen coupled to the frame buffer for receiving the plurality of image values read out from the frame buffer in the floating point format." Rossin fails to explicitly teach a processor for performing geometric calculations on a plurality of vertices of a primitive.

Deering teaches a floating point frame buffer such as the frame buffer 100 for storing floating point z-values (See Deering **column 14, lines 15-20** and column 16, lines 16-30) and "a display screen coupled to the frame buffer for receiving the plurality of image values read out from the frame buffer in the floating point format wherein Deering teaches rendering the floating-point z-values from the floating point frame buffer 100 (See Deering column 14, lines 15-20 and **column 17, lines 10-20**) and Deering teaches that the floating point z-values representing z coordinates for vertices of triangle primitives usable **to render** three-dimensional objects on display device 84 (column 10, lines 60-67 and column 11, lines 1-5).

Deering additionally discloses the claim limitation of a processor for performing geometric calculations on a plurality of vertices of a primitive wherein Deering teaches the floating point processors 152A-152F of Fig. 6 including the F-core block 202 for performing the

floating point intensive operations (which performs fixed point calculations as well) including the geometry transformation, clip testing, face determination, perspective division and screen space conversion (See column 7, lines 50-67).

Deering additionally discloses the claim limitation of a rasterization circuit coupled to the processor that rasterizes the primitive according to a rasterization process which operates using a floating point format wherein Deering teaches the draw processor 172A and 172B rasterizes the primitive (See column 9, lines 50-67 and column 10, lines 1-50) wherein the rasterization process includes partitioning primitives into pieces, sending a line description to the span field unit, performing an interpolation of values across arbitrarily oriented spans for triangles and lines, performing blending, anti-aliasing, depth cueing and setup for logical operations, etc. See Deering column 9, lines 50-67 and column 10, lines 1-50. The drawing processors 172A and 172B outputs floating point z-values to the floating point frame buffer 100 (See **column 14, lines 13-24**). Therefore, Deering's draw processor operates using a floating point format.

Therefore, having the combined teaching of Rossin, Deering and Olano as a whole, one of ordinary skill in the art would have found it obvious to modify the frame buffer of Rossin to achieve floating point precision (Olano Page 70) wherein *the floating point frame buffer generates a marked improvement in the rendered image quality* (Olano Fig. 4.7 and Page 59 and **Deering column 4, lines 25-30**) with more expensive computation load (Olano Page 69).

Moreover, Olano teaches a pixel processor for performing geometric calculations on a plurality of vertices of a primitive (See Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 68-75, 102-104). For example, Olano teaches pixel processor receives geometry primitive data and performs

either floating point or fixed point operations on the received geometry data. He discloses a graphics rendering pipeline mapped to the so called PixelFlow including a SIMD system of pixel processors performing modeling, transformation, primitive and interpolation, shading, lighting, atmospheric shading and image warping. In that the PixelFlow includes shaders for determining the shading and color variations across each surface wherein the shaders are executed sequentially including performing the surface shader to perform a certain class of texture lookups in which detailed surface geometry may be rendered using texture maps wherein the texture maps (Olano Page 31-32) are used to get different effects. In that he also teaches handling a floating point or fixed point frame buffer which is a portion of the rasterization pipeline within the graphics rendering pipeline. The color values received by the pipeline are represented in a floating point format which includes a mantissa portion and an exponent portion (Olano Page 100).

Fianlly, having the combined teaching of Rossin, Deering and Olano as a whole, one of ordinary skill in the art would have found it obvious to modify the rasterization process of Rossin which acts on the floating point color values that incorporates a processor in a graphics pipeline of Deering or Olano for performing geometric calculations on a plurality of vertices of a primitive. Doing so would enable the color values being represented more efficiently resulting in increased performance and accuracy (See Olano Fig. 4.7 and **Deering column 4, lines 25-30**) for the graphics pipeline (See Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 59, 68-79, 102-104) wherein the pipeline executing on the floating point values generates a marked improvement in the rendered image quality over the pipeline executing on the fixed point values (Olano Fig. 4.7 and

Page 59 and **Deering column 4, lines 25-30**) with more expensive computation load (Olano Page 69).

Re claims 2 and 46, Rossin and Olano disclose rasterization circuit performs scan conversion on vertices having floating point values (Rossin col. 2, lines 12-67). In other words, Rossin and Deering teach three-dimensional transformation, texture mapping, lighting, clipping, and perspective divide operations as well as plane equation generation performed in floating point format (Rossin col. 2, lines 12-67 and Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 68-79, 102-104; Deering Figs. 5-7).

Re claims 3 and 47, Deering and Olano disclose a texture circuit coupled to the rasterization circuit with the graphics pipeline that applies a texture to the primitive, wherein the texture is specified by floating point values and a texture memory coupled to the texture circuit that stores a plurality of textures in floating point values (See Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 68-75, 102-104 and Deering column 10, lines 20-50).

Re claims 5 and 48, Rossin, Deering and Olano disclose the floating point format is comprised of sixteen bits (Rossin col. 1, lines 32-44 and Deering Fig. 9A and 9C).

Rossin, Deering and Olano disclose floating point values have 16 bits (Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 68-79, 102-104 and Deering Figs. 9A and 9C)

Re claims 7 and 50, Rossin, Deering and Olano disclose a lighting circuit coupled to the rasterization circuit for performing a lighting function, wherein the lighting function executes on floating point values (Olano col. 2, lines 42-67 and Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 68-79, 102-104 and **Deering column 7, lines 50-67 and column 8, lines 1-23**).

Re claims 6, 8-13 and 22, 49, and 51-56, the limitations of claims 6, 8-13, 22, 49 and 51-56 are analyzed as discussed with respect to claim 1.

Re claim 26, Deering and Olano disclose the steps of writing, storing, and reading the data in the frame buffer in the floating point format are further comprised of specifying the floating point format according to a specification, wherein the specification corresponds to a level of range and precision (See Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 68-79, 102-104 and **Deering Fig 9 and column 14, lines 13-23**).

Re claim 31, Rossin, Deering and Olano disclose a computer system comprising a raster subsystem for performing a rasterization process, the rasterization process performed in a floating point format and a floating point frame buffer coupled to the raster subsystem for storing a plurality of floating point color values (Rossin col. 2, lines 12-67 and Olano Fig. 2.1, 3.1, 3.2, 3.4, 5.2 and Page 68-75, 102-104 and Deering Figs. 3-7). In other words, Rossin, Olano and Deering teach a typical computer graphics system include a raster subsystem in a graphics pipeline.

The output from the geometry accelerator, referred to as rendering data, is used by the rasterizer (and optional texture mapping hardware) to compute final screen space coordinates and R, G, B color values for each pixel constituting the primitives. The pixel data is stored in the frame buffer for display on a display screen. In that the geometry accelerator may be required to perform on the order of hundreds of millions of floating point calculations per second per chip. Functions of the geometry accelerator may include three-dimensional transformation, lighting, clipping, and perspective divide operations as well as plane equation generation, performed in

floating point format. Geometry accelerator functions result in rendering data which is sent to the frame buffer subsystem for rasterization.

Re claims 32-33 and 35, Deering and Olano disclose the floating point color values are written to, read from (for display purposes), and stored in the frame buffer (See Olano Fig. 2.1, 3.1, 3.2, 3.4, 4.7, 5.2 and Page 59, 68-79, 102-104 and **Deering column 10, lines 60-67 and column 11, lines 1-5**).

Re claims 36-37, Deering and Olano disclose the floating point color values are comprised of 16 bits of data and the data are comprised of one sign bit, ten mantissa bits, and five exponent bits (See Olano Fig. 2.1, 3.1, 3.2, 3.4, 4.7, 5.2 and Page 59, 68-79, 102-104 and Deering Fig. 9).

### *Conclusion*

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jin-Cheng Wang whose telephone number is (571) 272-7665. The examiner can normally be reached on 8:00 - 6:30 (Mon-Thu).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kee Tung can be reached on (571) 272-7794. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jcw